2º SEMESTRE: CÁLCULO NUMÉRICO CON FORTRAN.

TEMARIO:

1. Sistemas lineales. Directos e iterativos.
2. Cálculo de autovalores.
3. Derivación e integración numérica.
4. Solución de ecuaciones y sistemas no lineales.
5. Solución numérica de ecuaciones diferenciales.

FÓRMULAS DE DERIVACIÓN NUMÉRICA

Ideas básicas

Si una función f es al menos de clase C^2 en un entorno de un punto x, el teorema de Taylor nos permite escribir

$$f(x+h) = f(x) + \frac{f'(x)}{1!}h + \frac{f''(\xi_1)}{2!}h^2$$

para un cierto $\xi_1 \in (x, x+h)$. Si despejamos $f'(x)$

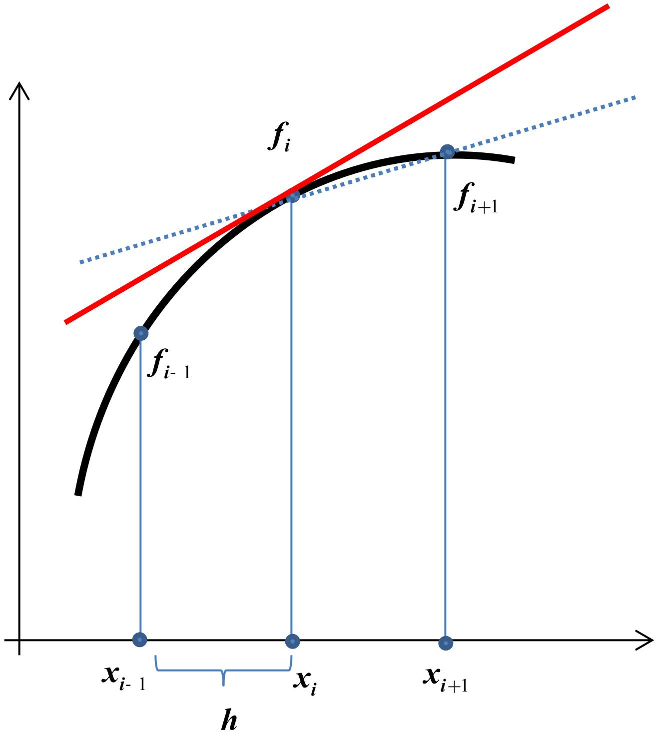
$$f'(x) = \frac{f(x+h) - f(x)}{h} - \frac{f''(\xi_1)}{2!}h$$

obtenemos una aproximación $O(h)$ para $f'(x)$

$$f'(x) \simeq \frac{f(x+h) - f(x)}{h}$$

FÓRMULAS DE DERIVACIÓN NUMÉRICA

Ideas básicas: Diferencias progresivas



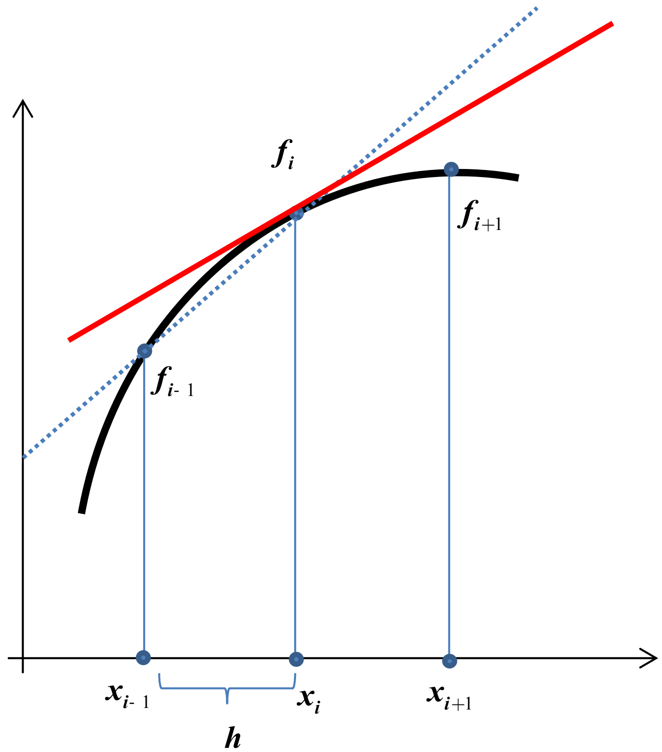
$$f_{i+1} = f_i + h\frac{f_i^{'}}{1!} + \underbrace{h^2\frac{f^{''}(\zeta)}{2!}}_{\text{Resto de Taylor}}$$

$$f_i^{'} = \frac{f_{i+1} - f_i}{h} + O(h)$$

Error de truncación: $O(h) = h\dfrac{f^{''}(\zeta)}{2!}$

FÓRMULAS DE DERIVACIÓN NUMÉRICA

Ideas básicas: Diferencias regresivas



$$f_{i-1} = f_i - h\frac{f_i^{'}}{1!} + \underbrace{h^2\frac{f^{''}(\zeta)}{2!}}_{\text{Resto de Taylor}}$$

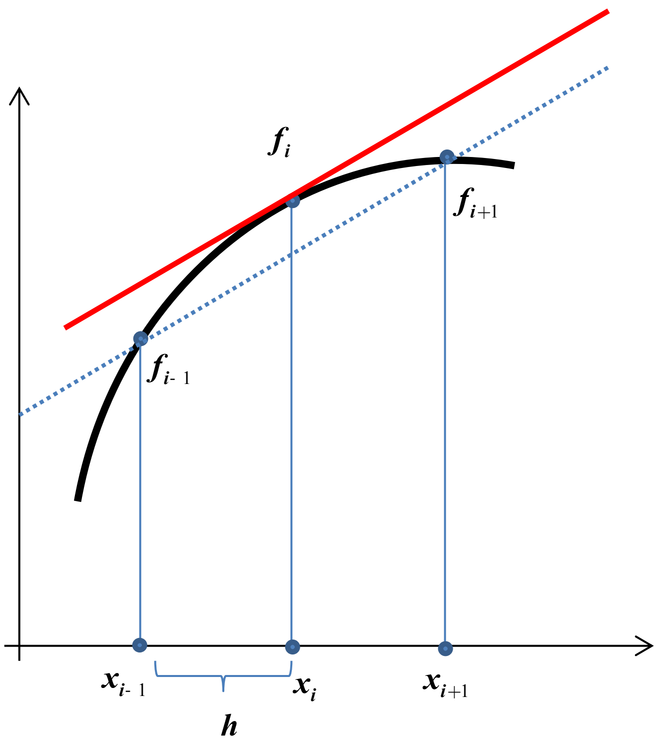
$$f_i^{'} = \frac{f_i - f_{i-1}}{h} + O(h)$$

$$\text{Error de truncación:} \quad O(h) = h\frac{f^{''}(\zeta)}{2!}$$

FÓRMULAS DE DERIVACIÓN NUMÉRICA

Ideas básicas: Diferencias centradas

- Se obtiene restando los dos resultados anteriores, lo que permite aumentar la precisión al eliminar el término de segundo grado: como utiliza más puntos disminuye el error de truncamiento



$$f_{i+1} = f_i + h\frac{f_i'}{1!} + h^2\frac{f''(\zeta)}{2!}$$

$$f_{i-1} = f_i - h\frac{f_i'}{1!} + h^2\frac{f''(\zeta)}{2!}$$

$$f_i' = \frac{f_{i+1} - f_{i-1}}{2h} + O(h^2)$$

$$\text{Error de truncación: } O(h^2) = h^2\frac{f'''(\zeta)}{6!}$$

FÓRMULAS DE DERIVACIÓN NUMÉRICA

Ideas básicas: Ordenes mayores

Diferencias progresivas:

Orden del método	
Primer orden $O(h)$	$f_i' = \dfrac{f_{i+1} - f_i}{h}$
	$f_i'' = \dfrac{f_{i+2} - 2f_{i+1} + f_i}{h^2}$
Segundo orden $O(h^2)$	$f_i' = \dfrac{-f_{i+2} + 4f_{i+1} - 3f_i}{2h}$
	$f_i'' = \dfrac{-f_{i+3} + 4f_{i+2} - 5f_{i+1} + 2f_i}{h^2}$

FÓRMULAS DE DERIVACIÓN NUMÉRICA

Ideas básicas: Ordenes mayores

Diferencias regresivas:

Orden del método	
Primer orden $O(h)$	$$f_i' = \frac{f_i - f_{i-1}}{h}$$
	$$f_i'' = \frac{f_{i-2} - 2f_{i-1} + f_i}{h^2}$$
Segundo orden $O(h^2)$	$$f_i' = \frac{f_{i-2} - 4f_{i-1} + 3f_i}{2h}$$
	$$f_i'' = \frac{-f_{i-3} + 4f_{i-2} - 5f_{i-1} + 2f_i}{h^2}$$

FÓRMULAS DE DERIVACIÓN NUMÉRICA

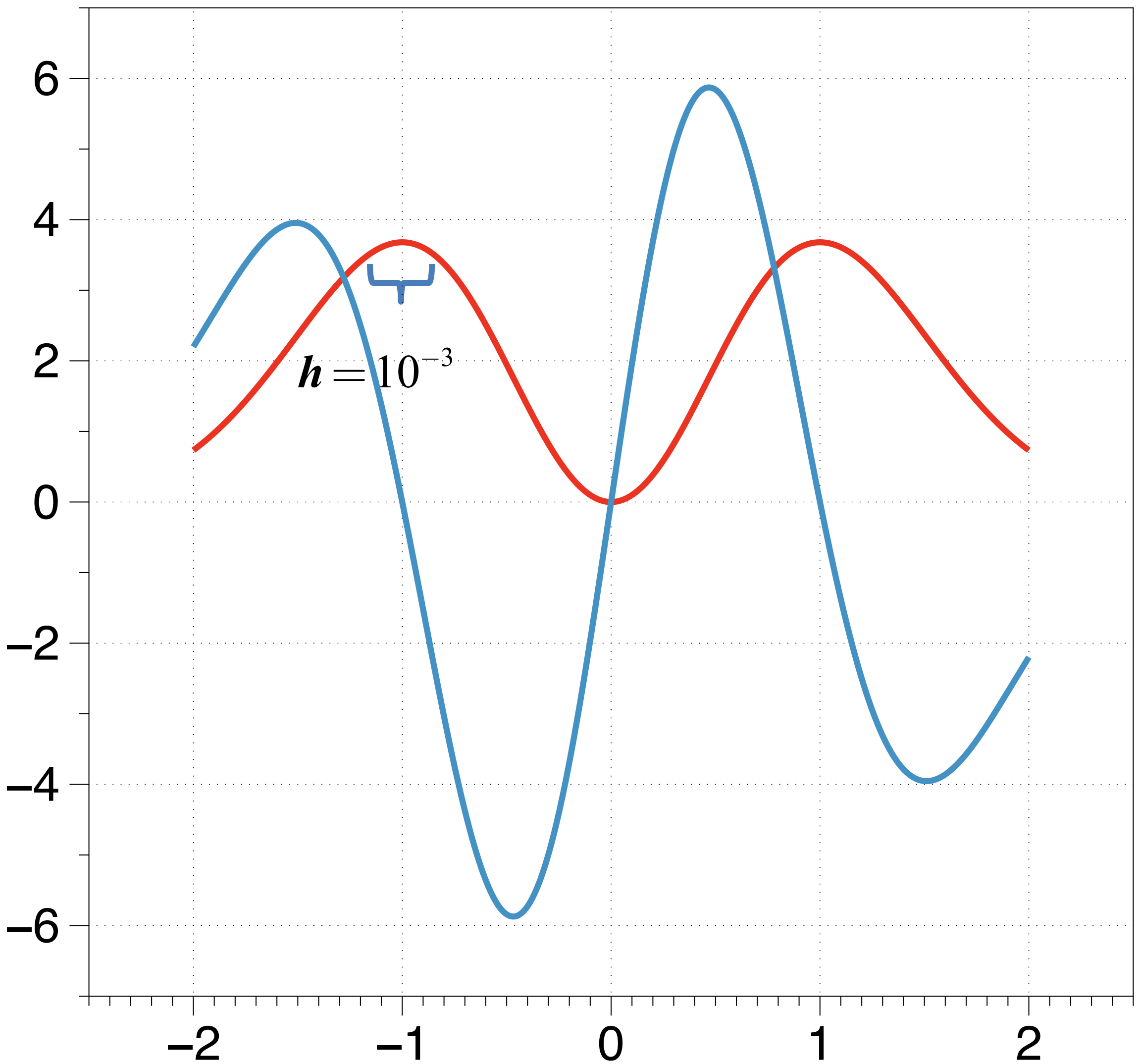
Ideas básicas: Ordenes mayores

Diferencias centradas:

Orden del método	
Primer orden $O\left(h^2\right)$	$f_i' = \dfrac{f_{i+1} - f_{i-1}}{2h}$
	$f_i'' = \dfrac{f_{i+1} - 2f_i + f_{i-1}}{h^2}$
Segundo orden $O\left(h^4\right)$	$f_i' = \dfrac{-f_{i+2} + 8f_{i+1} - 8f_{i-1} + f_{i-2}}{12h}$
	$f_i'' = \dfrac{-f_{i+2} + 16f_{i+1} - 30f_i + 16f_{i-1} - f_{i-2}}{12h^2}$

FÓRMULAS DE DERIVACIÓN NUMÉRICA

Ideas básicas: Ejemplo



$$\boxed{y = 10x^2 \cdot e^{-x^2}}$$

$$\boxed{y' = 20x \cdot e^{-x^2} - 20x^3 \cdot e^{-x^2}}$$

FÓRMULAS DE DERIVACIÓN NUMÉRICA

Práctica:

Función f(x)
Intervalo [a,b]
Tamaño paso h

→

Derivada f'(x)
Error cometido

```
PROGRAM PROGRAMA_QUE_DERIVA

    use DERIVACION_NUMERICA
    implicit none

    real(8) :: a, b, h

    !Bucle que calcule y pinte la derivada en los puntos

    !Los puntos estaran en el intervalo [a,b] a distancia h
    !Hay que pintar:
        !a) Valor de x
        !b) Valor de f(x)
        !c) Valor de df/dx utilizando
            !1.- Diferencias progresiva
            !2.- Diferencias regresiva
            !3.- Diferencias centradas
        !d) Error cometido por cada uno de los métodos
            !(Diferencia entre derivada numérica y derivada exacta)


END PROGRAM
```

FÓRMULAS DE DERIVACIÓN NUMÉRICA

Práctica:

Función f(x)
Intervalo [a,b]
Tamaño paso h

→

Derivada f'(x)
Error cometido

```
MODULE FUNCIONES

    implicit none

    contains

    function mifuncion(x)
        !Definicion de mi funcion
    end function

    function derivadademifuncion(x)
        !Definicion de la derivada de mi funcion
    end function


END MODULE FUNCIONES
```

FÓRMULAS DE DERIVACIÓN NUMÉRICA

Práctica:

Función f(x)
Intervalo [a,b]
Tamaño paso h

Derivada f'(x)
Error cometido

```
MODULE DERIVACION_NUMERICA

    implicit none

    contains

    function derivada_progresiva(f(x),x,h)
        !Funcion que calcula la derivada progresiva
    end function

    function derivada_regresiva(f(x),x,h)
        !Funcion que calcula la derivada regresiva
    end function

    function derivada_centrada(f(x),x,h)
        !Funcion que calcula la derivada centrada
    end function


END MODULE DERIVACION_NUMERICA
```

FÓRMULAS DE DERIVACIÓN NUMÉRICA

Práctica:

Función f(x)
Intervalo [a,b]
Tamaño paso h

→

Derivada f'(x)
Error cometido

Añadir función que calcule la derivada segunda:

1.- Utilizando una de las fórmulas anteriores
2.- Calculando la derivada segunda como:

$$\frac{d^2 f}{dx^2} = \frac{d}{dx}\left(\frac{df}{dx}\right)$$

Calculada numéricamente

3.- Calcular el error cometido en cada caso (diferencia con la solución exacta)

FÓRMULAS DE DERIVACIÓN NUMÉRICA

Complemento:

Función F(x,y)
Punto (x_0, y_0)
Tamaño paso h

$\Rightarrow$

Jacobiano J(x,y)
Error cometido

Supongamos $\mathbf{F}:\mathbb{R}^n\rightarrow\mathbb{R}^m$ es una función que va del espacio euclídeo *n*-dimensional a otro espacio euclídeo *m*-dimensional. Esta función está determinada por *m* funciones escalares reales:

$$y_i = F_i(x_1, \ldots, x_n), \qquad \mathbf{y} = \mathbf{F}(\mathbf{x}) = (F_1(\mathbf{x}), \ldots, F_m(\mathbf{x}))$$

Cuando la función anterior es diferenciable, entonces las derivadas parciales de estas *m* funciones pueden ser organizadas en una matriz *m* por *n*, la matriz jacobiana de **F**:

$$\begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_m}{\partial x_1} & \cdots & \frac{\partial y_m}{\partial x_n} \end{bmatrix}$$

Matriz Jacobiana

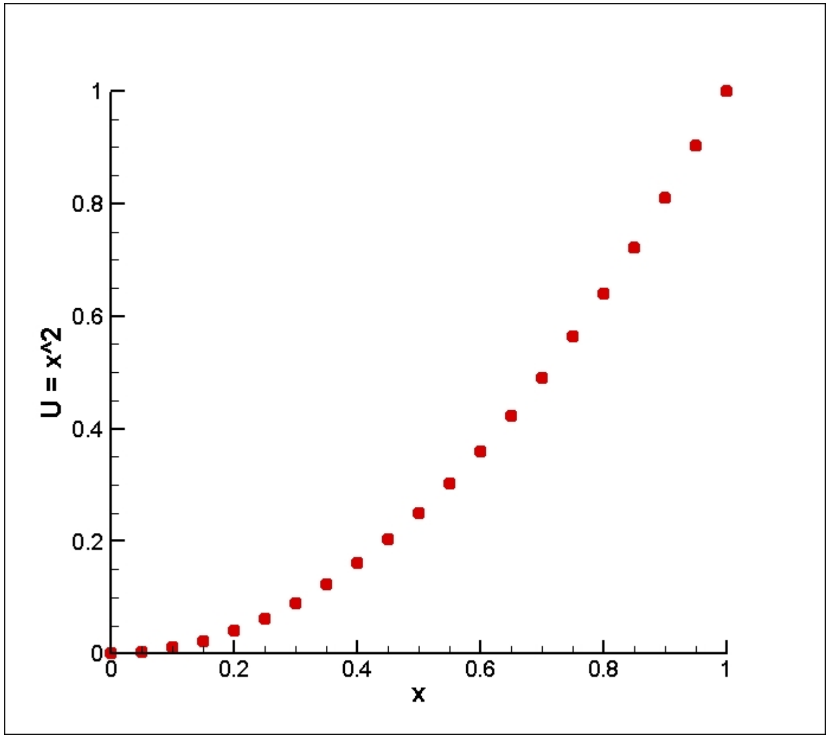
https://es.wikipedia.org/wiki/Jacobiano

Esta matriz es notada de diversas maneras:

$$J_{\mathbf{F}}(x_1, \ldots, x_n), \qquad \text{o} \qquad \frac{\partial(y_1, \ldots, y_m)}{\partial(x_1, \ldots, x_n)}, \qquad \text{o} \qquad D\mathbf{F}(x_1, \ldots, x_n),$$

PROBLEMA DE CONTORNO

✓ Se basa en reescribir el problema diferencial en forma matricial y usar alguno de los algoritmos vistos para resolver problemas lineales.



$$x_i = \{0{,}0.05{,}0.1{,}0.15 \ldots 0.95{,}1\}$$

$$U(x_i) = x_i^2$$

Usando diferencias centradas:

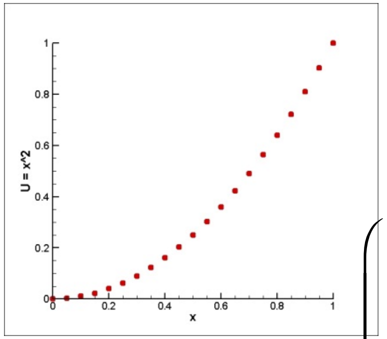
$$U'(x_i) = \frac{U(x_{i+1}) - U(x_{i-1})}{2 \cdot \Delta x}$$

Construimos la matriz de derivación

$$D \ U(x_i) = U'(x_i)$$

PROBLEMA DE CONTORNO

- ✓ Se basa en reescribir el problema diferencial en forma matricial y usar alguno de los algoritmos vistos para resolver problemas lineales.

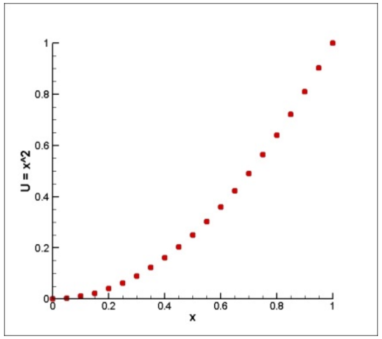


$$D\ U(x_i) = U'(x_i)$$

$$\begin{pmatrix} -\frac{1}{\Delta x} & \frac{1}{\Delta x} & & & & \\ \frac{-1}{2\Delta x} & 0 & \frac{1}{2\Delta x} & & & \\ & \frac{-1}{2\Delta x} & 0 & \frac{1}{2\Delta x} & & \\ & & \ddots & \ddots & \ddots & \\ & & & \frac{-1}{2\Delta x} & 0 & \frac{1}{2\Delta x} \\ & & & & -\frac{1}{\Delta x} & \frac{1}{\Delta x} \end{pmatrix} \begin{pmatrix} U(x_0) \\ U(x_1) \\ U(x_2) \\ \vdots \\ U(x_{n-1}) \\ U(x_n) \end{pmatrix} = \begin{pmatrix} U'(x_0) \\ U'(x_1) \\ U'(x_2) \\ \vdots \\ U'(x_{n-1}) \\ U'(x_n) \end{pmatrix}$$

PROBLEMA DE CONTORNO

- ✓ Se basa en reescribir el problema diferencial en forma matricial y usar alguno de los algoritmos vistos para resolver problemas lineales.



$$D\ U(x_i) = U'(x_i)$$

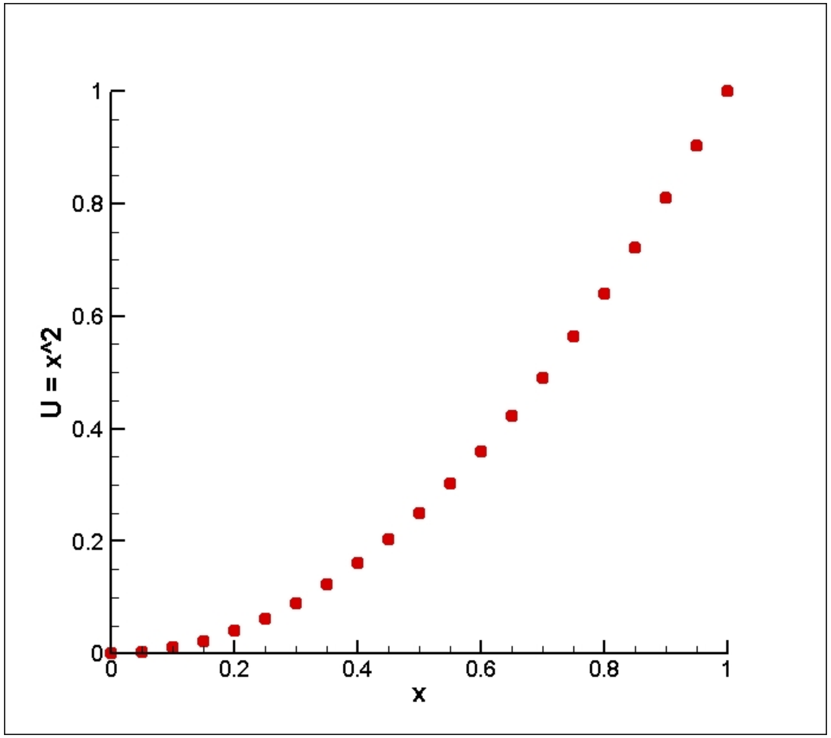
$$U'(x_1) = \frac{U(x_2) - U(x_0)}{2 \cdot \Delta x}$$

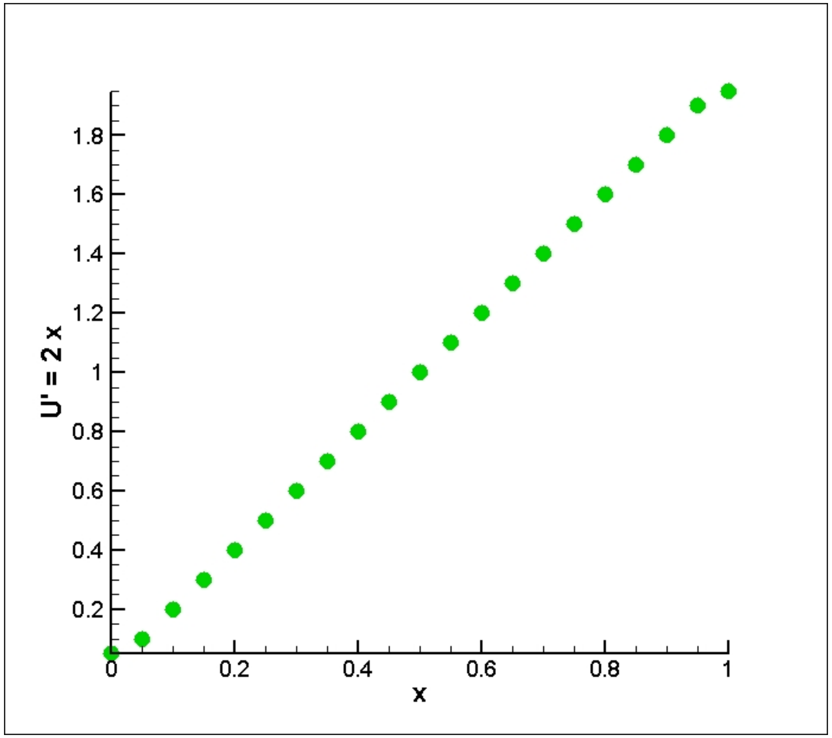
$$\begin{pmatrix} -\frac{1}{\Delta x} & \frac{1}{\Delta x} & & & & \\ \boxed{\frac{-1}{2\Delta x} \quad 0 \quad \frac{1}{2\Delta x}} & & & & & \\ & \frac{-1}{2\Delta x} & 0 & \frac{1}{2\Delta x} & & \\ & & \ddots & \ddots & \ddots & \\ & & & \frac{-1}{2\Delta x} & 0 & \frac{1}{2\Delta x} \\ & & & & -\frac{1}{\Delta x} & \frac{1}{\Delta x} \end{pmatrix} \boxed{\begin{pmatrix} U(x_0) \\ U(x_1) \\ U(x_2) \\ \vdots \\ U(x_{n-1}) \\ U(x_n) \end{pmatrix}} = \begin{pmatrix} U'(x_0) \\ \boxed{U'(x_1)} \\ U'(x_2) \\ \vdots \\ U'(x_{n-1}) \\ U'(x_n) \end{pmatrix}$$

PROBLEMA DE CONTORNO

- ✓ Se basa en reescribir el problema diferencial en forma matricial y usar alguno de los algoritmos vistos para resolver problemas lineales.

$$D\ U(x_i) = U'(x_i)$$

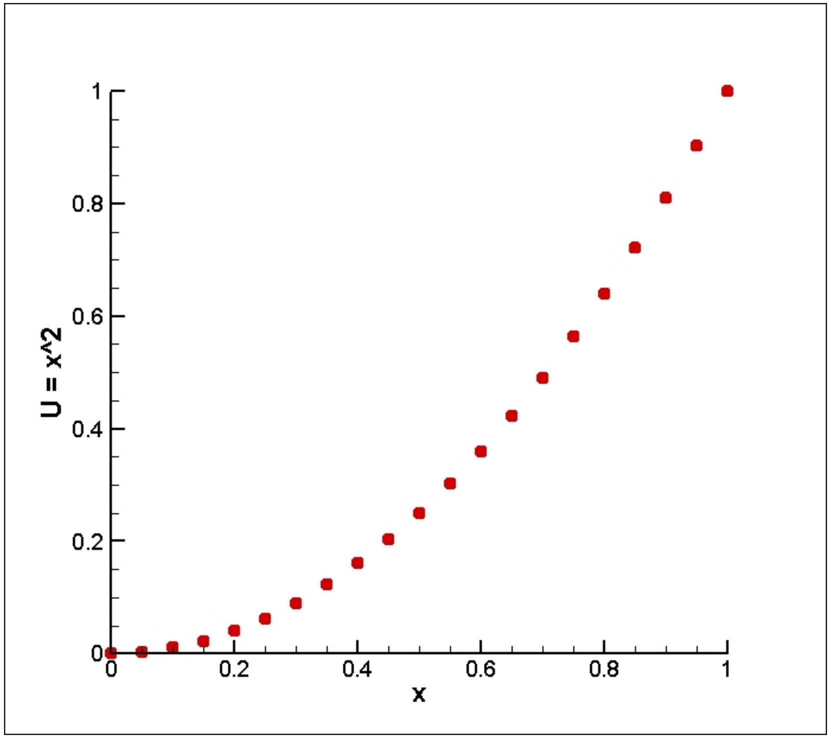


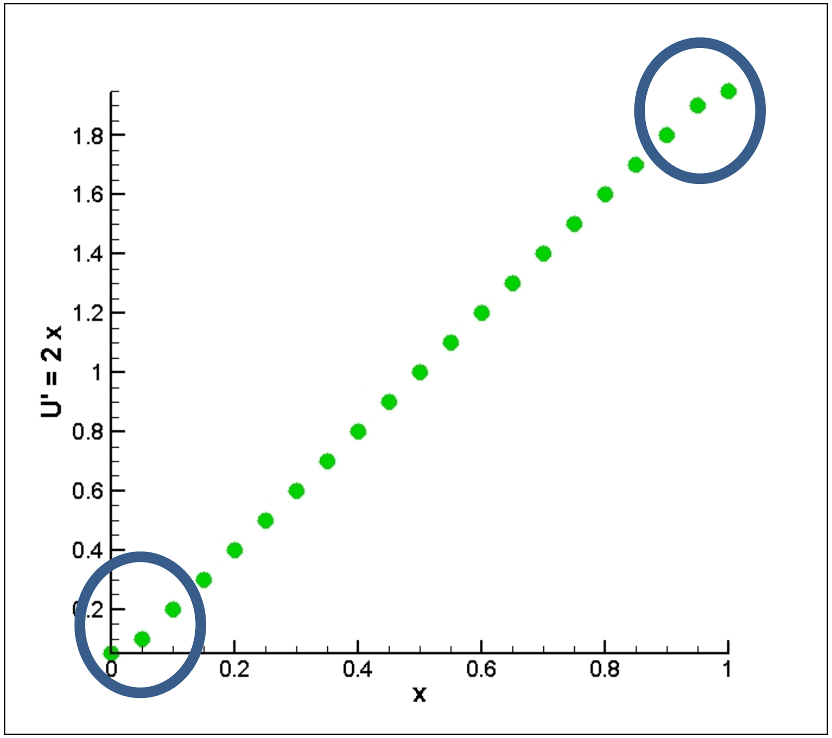


PROBLEMA DE CONTORNO

- ✓ Se basa en reescribir el problema diferencial en forma matricial y usar alguno de los algoritmos vistos para resolver problemas lineales.

$$D\ U(x_i) = U'(x_i)$$





PROBLEMA DE CONTORNO

- Se basa en reescribir el problema diferencial en forma matricial y usar alguno de los algoritmos vistos para resolver problemas lineales.

- Podemos usarlo para resolver el problema que usamos como ejemplo

$$\begin{cases} U''(x) + U(x) = 0 \\ U(0) = 1; U(\pi/2) = 0 \end{cases}$$

- Discretizamos el intervalo

$$x_0 = 0 \qquad\qquad x_n = \frac{\pi}{2}$$

PROBLEMA DE CONTORNO

- Reescribiendo la ecuación diferencial en forma matricial

$$\mathrm{U}''(x)+\mathrm{U}(x)=0 \Rightarrow M(U)=f(x)$$

- En este caso:

$$M\big(\mathrm{U}(x)\big)=0 \text{ con } M=D\cdot D+I$$

siendo D la matriz de derivación anterior

$$M=\begin{pmatrix} \ddots & & \\ & D & \\ & & \ddots \end{pmatrix}\cdot\begin{pmatrix} \ddots & & \\ & D & \\ & & \ddots \end{pmatrix}+\begin{pmatrix} \ddots & & \\ & 1 & \\ & & \ddots \end{pmatrix}$$

Basta imponer los contornos:

PROBLEMA DE CONTORNO

Basta imponer los contornos: $\begin{cases} \mathrm{U}''(x)+\mathrm{U}(x)=0 \\ \mathrm{U}(0)=1; U(\pi/2)=0 \end{cases}$

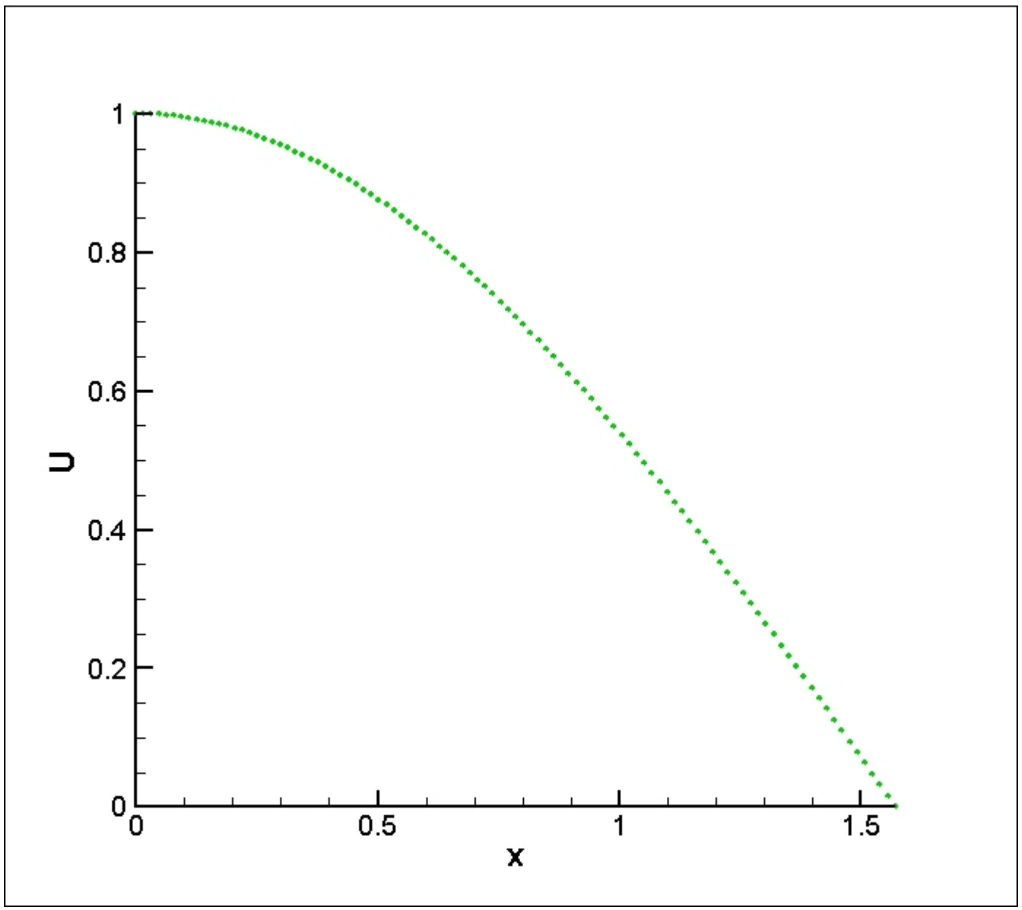
$$\begin{pmatrix} \boxed{1 \quad 0 \quad .. \quad . \quad 0} \\ M_{10} \quad . \quad . \quad . \quad M_{1n} \\ . \quad . \quad . \quad . \quad . \\ M_{n-10} \quad . \quad . \quad . \quad M_{n-1n} \\ \boxed{0 \quad . \quad . \quad . \quad 1} \end{pmatrix} \begin{pmatrix} U(x_0) \\ U(x_1) \\ . \\ . \\ U(x_n) \end{pmatrix} = \begin{pmatrix} 1 \\ F(x_1) \\ . \\ F(x_{n-1}) \\ 0 \end{pmatrix} \Rightarrow \begin{matrix} \mathrm{U}(x_0)=\mathrm{U}(0)=1 \\ \\ \mathrm{U}(x_n)=\mathrm{U}(\pi/2)=0 \end{matrix}$$

Queda un problema del tipo $Ax=b$ donde el vector de incógnitas está formado por el valor de la función desconocida en los puntos de la malla.

PROBLEMA DE CONTORNO

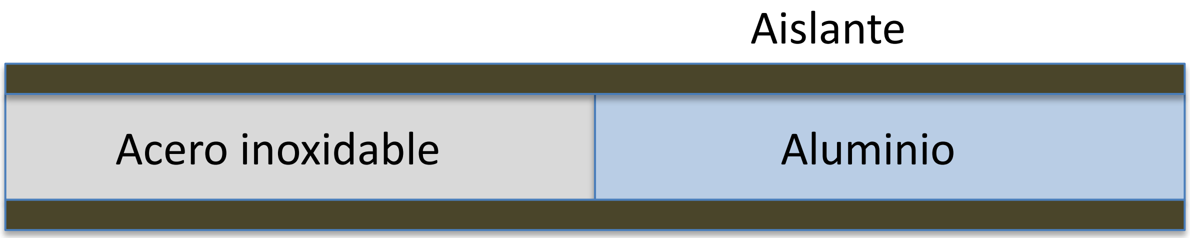
Basta imponer los contornos: $\begin{cases} U''(x) + U(x) = 0 \\ U(0) = 1; U(\pi/2) = 0 \end{cases}$

Queda un problema del tipo $Ax = b$ donde el vector de incógnitas está formado por el valor de la función desconocida en los puntos de la malla.

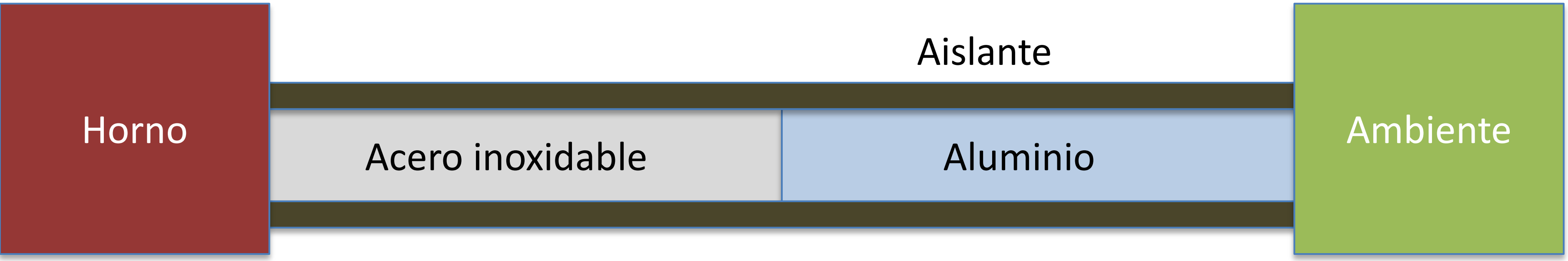


PROBLEMA DE CONTORNO: Ecuación del calor 1D

Tenemos dos barras, de 0.5 m cada una, en contacto. Las barras están recubiertas de material aislante y abiertas en los extremos izquierdo y derecho.



Introducimos el extremo izquierdo en un horno a 500K, mientras que el derecho puede considerarse que siempre se encuentra a temperatura ambiente (298.15K)



¿Cuál es la distribución de temperatura final de la barra T(x)?

Nota:

Acero inoxidable: k=16.3 W/mK Aluminio: k=209.3W/mK

PROBLEMA DE CONTORNO: Ecuación del calor 1D

La ecuación del calor unidimensional es:

$$\rho c_p \frac{\partial T}{\partial t} - \frac{\partial}{\partial x}\left(k(x) \frac{\partial T}{\partial x} \right) = 0$$

- k is thermal conductivity (W/(m·K))
- ρ is density (kg/m³)
- c_p is specific heat capacity (J/(kg·K))

La solución estacionaria (derivada temporal cero) se obtiene de resolver el siguiente problema:

$$\frac{\partial}{\partial x}\left(k(x) \frac{\partial T}{\partial x} \right) = 0$$

Con un conjunto adecuado de condiciones de contorno.

Un conjunto adecuado de condiciones de contorno para este problema son los valores de temperatura en los extremos del dominio en el que queramos resolver la ecuación.

PROBLEMA DE CONTORNO: Ecuación del calor 1D

Problema de contorno basado en la ecuación del calor 1D

$$\frac{\partial}{\partial x}\left(k(x)\frac{\partial T}{\partial x}\right)=0 \qquad x=[a,b]$$

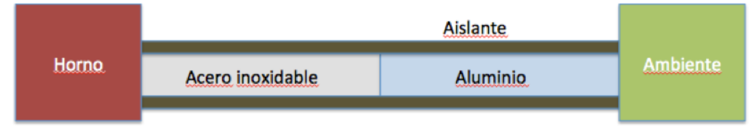
$$T(a)=T_0 \quad T(b)=T_1$$

- k is thermal conductivity (W/(m·K))
- ρ is density (kg/m³)
- c_p is specific heat capacity (J/(kg·K))

Este es un modelo matemático adecuado para modelizar el problema de la barra.

La discretización del problema utilizando la técnica de diferencias finitas, nos permite encontrar una solución aproximada del problema matemático.

PROBLEMA DE CONTORNO: Ecuación del calor 1D



1.- Resolver el problema de calor unidimensional de las dos barras.

2.- Parte del problema es resolver un sistema de tipo Ax=b. Para ello:

a) Comprobar, calculando el radio espectral, si se pueden usar los métodos de Jacobi o Gauss Seidel. NOTA: el código debe ser robusto, y el método de la potencia puede no funcionar. Si no funciona, ¿diagonal dominante?

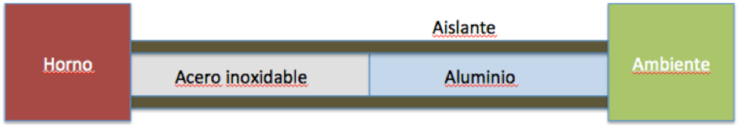
b) En caso de que la convergencia esté garantizada, usar el método iterativo.

c) En caso contrario usar LU sin pivote.

d) Si es necesario el pivote usar Gauss con pivote.

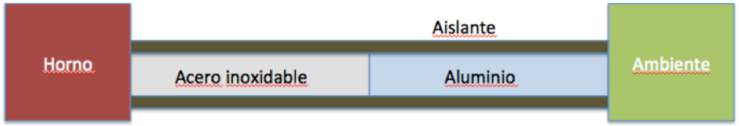
Entregar proyecto (archivos *.f90 junto con Makefile) que resuelva el problema de las dos barras. Adjuntar un informe en PDF de 400-500 palabras explicando cómo se ha resuelto el problema.

PROBLEMA DE CONTORNO: Ecuación del calor 1D

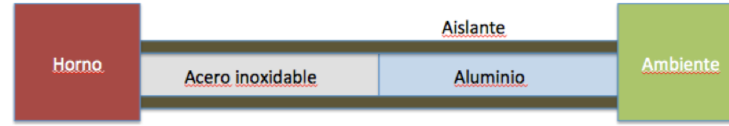


- Evaluación:
  - Aprobado: solución del problema con método de Gauss con pivote.
  - Bien: solución del problema con método iterativo (solo en caso de que sea posible). Si no es posible, por Gauss con pivote.
  - Notable: solución del problema con método iterativo (solo en caso de que sea posible). Si no es posible, por LU sin pivote. Si no es posible, por Gauss con pivote.
  - Sobresaliente: Ver puntos extras.
  - MH: Ver puntos extras.

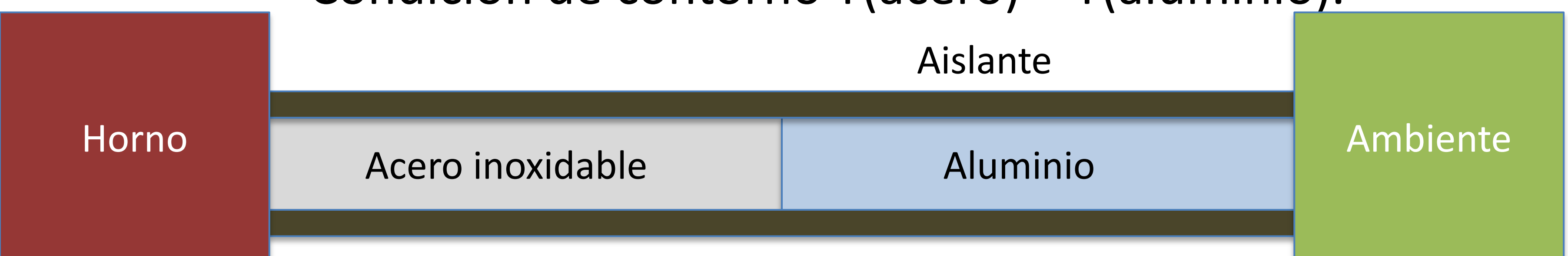
PROBLEMA DE CONTORNO: Ecuación del calor 1D



- El informe debe incluir la respuesta a las siguientes cuestiones:
  - Solución del problema (gráfica T(x)).
  - Explicación de la gráfica anterior.
  - Error en función del número de puntos para distintos órdenes de derivación (gráfica E(N)).
  - Explicación de la gráfica anterior.



- Evaluación:
  - Punto extra 1:
    - Si consideráis k(x) como una función discontinua la solución presentará oscilaciones. Arreglar el problema considerando dos problemas de contorno acoplados. Condición de contorno T(acero) = T(aluminio).

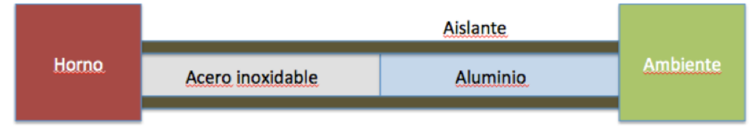


¿Cuál es la distribución de temperatura final de la barra T(x)?

Nota:

Acero inoxidable: k=16.3 W/mK   Aluminio: k=209.3W/mK

PROBLEMA DE CONTORNO: Ecuación del calor 1D



- Evaluación:
  - Punto extra 2:
    - Si el informe está especialmente bien escrito, se asignará un punto extra al trabajo.

La semana que viene publicaré un F.A.Q. del problema.

Entrega por Moodle. Una entrega por grupo.
Nombre del fichero: Grupo_XX.zip
Fecha de entrega: 01/04/2018